

Exercícios de revisão

Versão: 20 de janeiro de 2022

Considere o seguinte:

- ❖ Um comboio tem **3** carruagens, com carga máxima de **c** passageiros, e dentro de cada carruagem tem **p** passageiros.
- ❖ Cada carruagem tem uma classe associada: primeira, segunda e terceira classe.
- ❖ A primeira classe tem uma carga máxima de 5 passageiros, a segunda classe tem uma carga máxima de 8 passageiros e a terceira classe tem uma carga máxima de 10 passageiros.
- ❖ Cada passageiro tem um bilhete (único) e este é intransmissível, e é representado da seguinte forma: (numero_bilhete, nome_passageiro, idade_do_passageiro)

O comboio pode ser representado da seguinte forma:

```
comboio = {  
    "primeira-classe": (5, [(12345, 'Anabela', 19), (60032, 'Ricardo', 30), (43809, 'João', 78)]),  
    "segunda-classe": ...,  
    "terceira-classe": ...  
}
```

Na primeira classe, a carga máxima são 5 passageiros, e tem de momento 3 passageiros: a Anabela com o bilhete 12345 e com idade 19 anos, o Ricardo e o João.

1. Crie a função `comboio_pronto_para_viagem(c_1classe, c_2classe, c_3classe)` que cria o dicionário que representa o comboio quando este se encontra vazio e pronto a partir. 'c_1classe' corresponde à carga máxima da primeira classe e assim sucessivamente.
2. Crie a função `entrar_no_comboio(comboio, nome, bilhete, idade, classe)` que permita adicionar um passageiro ao comboio.
3. Crie a função `entrar_no_comboio_ficheiro(nome_do_ficheiro)` que lê o ficheiro 'comboio.csv' e adiciona todos os passageiros ao comboio. Nota: deve usar a função definida em 2.
 - 3.1. Se o ficheiro não existir deverá tratar a exceção, e deverá imprimir uma mensagem ao utilizador a indicar que este não existe.
4. Crie a função `sair_do_comboio(comboio, bilhete)` que permita remover um passageiro do comboio.
5. Crie a função `procurar_passageiro(comboio, bilhete)` que permita procurar um passageiro pelo nº do bilhete.
6. Crie a função `procurar_passageiro_nome(comboio, nome)` que permita procurar um passageiro pelo nome.
 - 6.1. Se houver mais do que um nome que contenha o nome passado, deve retornar todos os que encontrar.
7. Crie a função `lotacao_carruagem(comboio, classe)` que devolva a lotação em percentagem preenchida na carruagem da classe.
8. Crie a função `lotacao_comboio(comboio)` que devolva a lotação em percentagem preenchida do comboio.
9. Crie a função `quantidade_seniores(comboio)` que devolva o número de pessoas com idade superior a 65 anos em todo o comboio.
10. Ordene os passageiros em cada carruagem pela idade. O critério de desempate será a ordem alfabética. Use a função `sorted` e expressões `lambda`.